CSE 142 Section Handout #4

Challenge Sheet

You are not expected or required to solve these problems. These problems are designed for students who want a fun, extra challenge to test their skills on harder programming problems. Have fun!

Pascal's Triangle is a famous triangle arrangement of numbers that has a lot of interesting mathematical properties. Pascal's Triangle starts out with the top two rows: 1 and 1 1. Then each row after that starts and ends with a 1 while every other number in the row is a sum of the two numbers above it. Note that this can go on infinitely and Pascal's Triangle can have an arbitrary number of rows. An example of Pascal's Triangle up to the 5th row is shown below on the left.

One of the more useful properties of Pascal's Triangle is its connection to binomial coefficients. It turns out you can calculate any number in Pascal's Triangle by using nCr where n is the row number and r is the element in that row (nCr is shorthand for "n Choose r" which some of you may recognize from combinations). You can use this knowledge to construct Pascal's Triangle as seen below on the right.

The formula for nCr is: $\dfrac{n!}{(n-r)!\,r!}$ and is often written as $\binom{n}{r}$

Note that r! is just shorthand for r × (r -1) × …. × 2 × 1. For example, 4! Is just 4 × 3 × 2 × 1

$$
\begin{array}{lccccccccccc}
n=0 & & & & & & 1 & & & & & \\
n=1 & & & & & 1 & & 1 & & & & \\
n=2 & & & & 1 & & 2 & & 1 & & & \\
n=3 & & & 1 & & 3 & & 3 & & 1 & & \\
n=4 & & 1 & & 4 & & 6 & & 4 & & 1 & \\
n=5 & 1 & & 5 & & 10 & & 10 & & 5 & & 1
\end{array}
\qquad
\begin{array}{ccccccccccc}
 & & & & & \binom{0}{0} & & & & & \\
 & & & & \binom{1}{0} & & \binom{1}{1} & & & & \\
 & & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & & \\
 & & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & & \\
 & \binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4} & \\
\binom{5}{0} & & \binom{5}{1} & & \binom{5}{2} & & \binom{5}{3} & & \binom{5}{4} & & \binom{5}{5}
\end{array}
$$

Write a method named pascalsTriangle that takes in an integer as a parameter and prints out that number of rows of Pascal's Triangle. For example, pascalsTriangle(6) would produce the following:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

You may notice once you're done with this method, that you start getting weird values with parameter inputs of more than 13. This is normal and something to do with how integers work in Java. It should only need to work with input values up to 13.

CSE 142 Section Handout #4

Solutions (Two Possible Solutions Provided)

1.

```
public static void pascalsTriangle(int numberofRows) {
  for (int row = 0; row < numberofRows; row++) {
    for (int element = 0; element <= row; element++) {
      System.out.print(combination(row, element) + " ");
    }
    System.out.println();
  }
}

public static int combination(int n, int r) {
  int nFactorial = 1;
  int rFactorial = 1;
  int nMinusRFactorial = 1;
  for (int i = 1; i <= n; i++) {
    nFactorial *= i;
  }
  for (int i = 1; i <= r; i++) {
    rFactorial *= i;
  }
  for (int i = 1; i <= n - r; i++) {
    nMinusRFactorial *= i;
  }
  return nFactorial / (rFactorial * nMinusRFactorial);
}
```

2.

```
public static void pascalsTriangle(int numberofRows) {
  for (int row = 1; row <= numberofRows; row++) {
    int element = 1;
    for (int index = 1; index <= row; index++) {
      System.out.print(element + " ");
      element *= (row - index);
      element /= index;
    }
    System.out.println();
  }
}
```